JUNIT - EXECUTION PROCEDURE

http://www.tutorialspoint.com/junit/junit_execution_procedure.htm

This tutorial explains the execution procedure of methods in JUnit which means that which method is called first and which one after that. Here is the execution procedure of the JUint test API methods with the example.

Create a java class file name JunitAnnotation.java in **C:\ > JUNIT_WORKSPACE** to test annotation

```
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;

public class ExecutionProcedureJunit {

   //execute only once, in the starting
   @BeforeClass
   public static void beforeClass() {
      System.out.println("in before class");
   }

   //execute only once, in the end
   @AfterClass
   public static void  afterClass() {
      System.out.println("in after class");
   }

   //execute for each test, before executing test
   @Before
   public void before() {
      System.out.println("in before");
   }

   //execute for each test, after executing test
   @After
   public void after() {
      System.out.println("in after");
   }

   //test case 1
   @Test
   public void testCase1() {
      System.out.println("in test case 1");
   }

   //test case 2
   @Test
   public void testCase2() {
      System.out.println("in test case 2");
   }
}
```

Next, let's create a java class file name **TestRunner.java** in **C:\ > JUNIT_WORKSPACE** to execute annotaions

```
import org.junit.runner.JUnitCore;
import org.junit.runner.Result;
import org.junit.runner.notification.Failure;

public class TestRunner {
   public static void main(String[] args) {
      Result result = JUnitCore.runClasses(ExecutionProcedureJunit.class);
      for (Failure failure : result.getFailures()) {
         System.out.println(failure.toString());
      }
```

```
        System.out.println(result.wasSuccessful());
    }
}
```

Compile the Test case and Test Runner classes using javac

```
C:\JUNIT_WORKSPACE>javac ExecutionProcedureJunit.java TestRunner.java
```

Now run the Test Runner which will run test case defined in provided Test Case class.

```
C:\JUNIT_WORKSPACE>java TestRunner
```

Verify the output.

```
in before class
in before
in test case 1
in after
in before
in test case 2
in after
in after class
```

See the above output and this is how the JUnit execution procedure is.

- First of all beforeClass() method execute only once
- Lastly, the afterClass() method executes only once.
- before() method executes for each test case but before executing the test case.
- after() method executes for each test case but after the execution of test case
- In between before() and after() each test case executes.